The Sparks Foundation GRIP Task - (1)

Prediction Using Supervised Machine Learning

Author- Anshuman Agarwal

AIM - Predict the score of an student based on the number of study hours. What will be the predicted score if a student studies for 9.25 hours per day ?

Importing Relevent Libraries

In [2]:

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from statsmodels.formula.api import ols
import scipy
from scipy import stats
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
```

Importing Data

In [3]:

```
data= pd.read_csv("https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student
```

In [4]:

```
print(data)
```

```
    Hours  Scores
0     2.5      21
1     5.1      47
2     3.2      27
3     8.5      75
4     3.5      30
5     1.5      20
6     9.2      88
7     5.5      60
8     8.3      81
9     2.7      25
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
15    8.9      95
16    2.5      30
17    1.9      24
18    6.1      67
19    7.4      69
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86
```

In [5]:

```
data.shape                                  # Shape represented in form of (rows,colum
```

Out[5]:

```
(25, 2)
```

Data Preprocessing

In [6]:

```
data.isnull().sum()                         # To check the count of missing values pre
```

Out[6]:

```
Hours     0
Scores    0
dtype: int64
```

There is no missing value in the given data set as each variables are showing 0 value in front of them.

In [7]:

```
data.describe()                                    # Few description about the data set.
```

Out[7]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

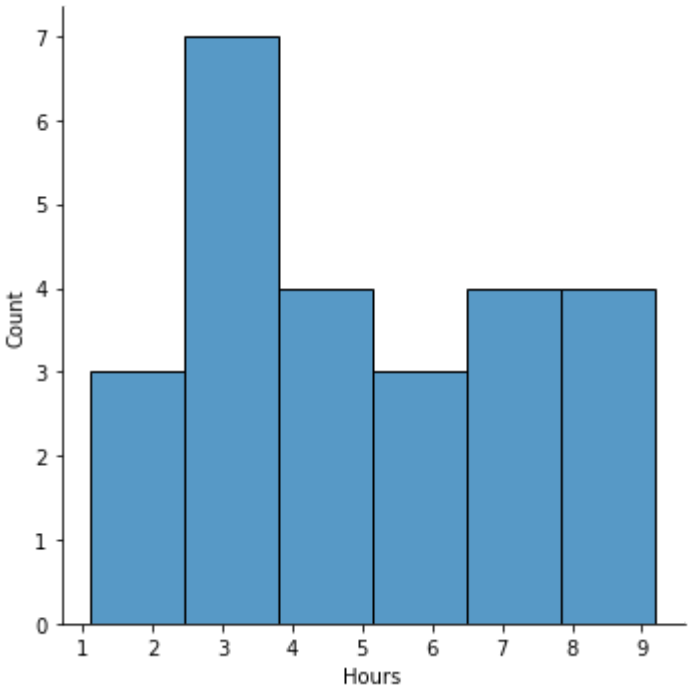
Data Distribution Plot

In [8]:

```
sns.displot(data['Hours'])                                    # Checking frequency distribut
```

Out[8]:

```
<seaborn.axisgrid.FacetGrid at 0x1acbc922790>
```




```
sns.displot(data['Scores'])
```

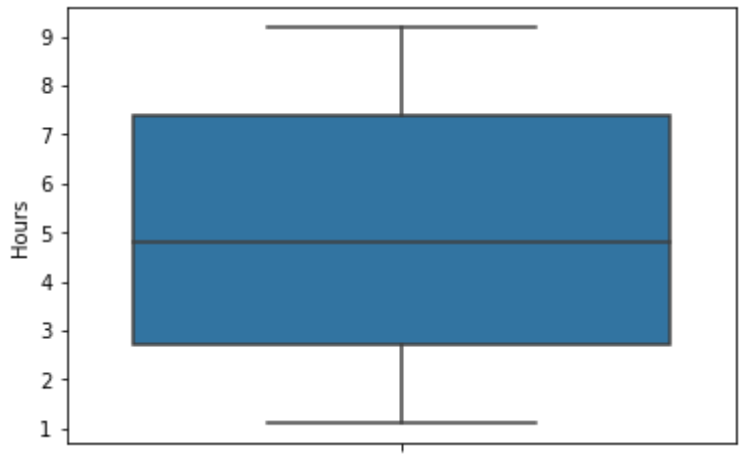
Box and whiskers plot of variables "Hours" & "Scores" to visually interpret the five-number summary (mean, median, 1st quantile, 2nd quantile, 3rd quantile) and tells about outliers present in the data.

In [10]:

```
sns.boxplot(y= data['Hours'])                    # Checking for the outliers.
```

Out[10]:

```
<AxesSubplot:ylabel='Hours'>
```

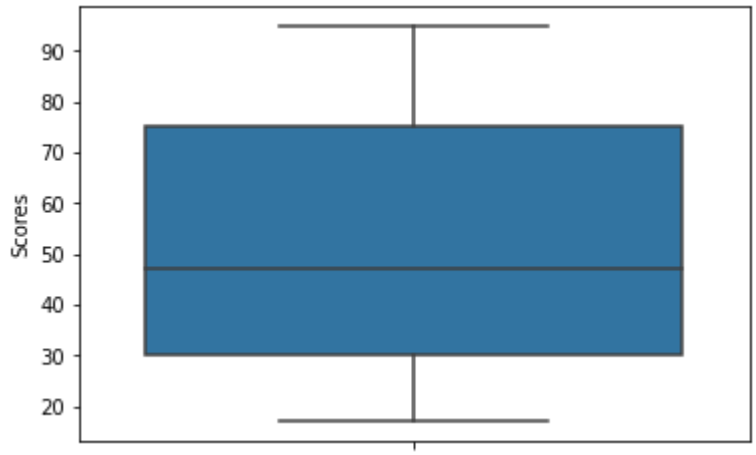


In [11]:

```
sns.boxplot(y= data['Scores'])
```

Out[11]:

```
<AxesSubplot:ylabel='Scores'>
```

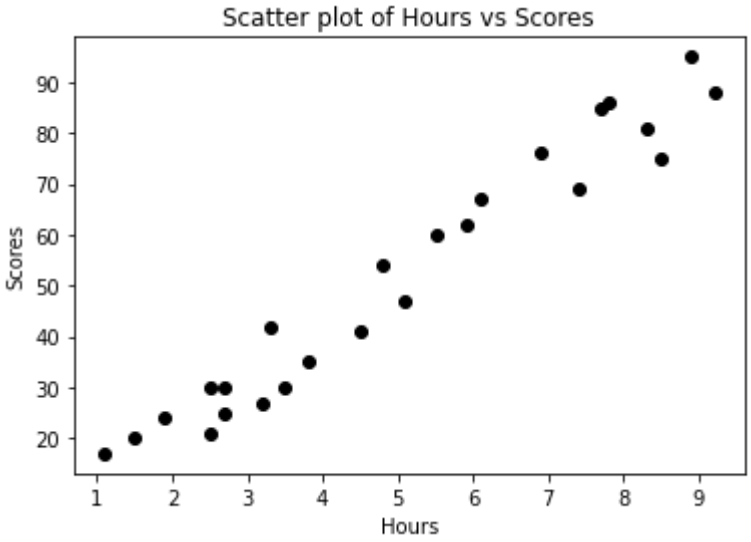


There is no sign of any outlier present in the data set.

Data Visualization

In [12]:

```
plt.scatter(data['Hours'],data['Scores'], c = 'black')
plt.title('Scatter plot of Hours vs Scores')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```



The scatter plot shows that there exist a positive linear relationship between score of student and hours of study. Thus linear function would be a correct function for the model.

In [13]:

```
data.corr()                                    # correlation between numerical
```

Out[13]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

The correlation also tells us that there is a high positive correlation (i.e 0.9761) between the variable "Hours" and "Scores".

Specifying the Model

In [27]:

```
x = data['Hours']                              # Defining the dependent variable and indep
y = data['Scores']
```

In [28]:

```
x=data['Hours'].values.reshape(-1,1)
y=data['Scores'].values.reshape(-1,1)
from sklearn.model_selection import train_test_split          # Splitting the data set into

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=56)
```

In [29]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[29]:

```
((20, 1), (5, 1), (20, 1), (5, 1))
```

In [30]:

```
len(x_train)
```

Out[30]:

```
20
```

In [31]:

```
x_train
```

Out[31]:

```
array([[3.3],
       [4.5],
       [9.2],
       [1.5],
       [7.8],
       [2.7],
       [8.3],
       [5.1],
       [5.5],
       [7.7],
       [1.9],
       [6.9],
       [2.7],
       [1.1],
       [5.9],
       [3.2],
       [2.5],
       [8.9],
       [3.5],
       [4.8]])
```

In [32]:

```
len(x_test)
```

Out[32]:

```
5
```

In [43]:

```
from sklearn.linear_model import LinearRegression    # Training the simple linear Regressio
regressor =LinearRegression()
regressor.fit(x_train,y_train)
```

Out[43]:

```
LinearRegression()
```

In [44]:

```
regressor.coef_
```

Out[44]:

```
array([[10.17148086]])
```

In [45]:

```
regressor.intercept_
```

Out[45]:

```
array([1.21831782])
```

In [46]:

```
x=sm.add_constant(x_train)
linear_regression_model = sm.OLS(y_train,x).fit()                    # Fitting the linear
linear_regression_model.summary()
```

Out[46]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.963
Model:	OLS	**Adj. R-squared:**	0.961
Method:	Least Squares	**F-statistic:**	474.5
Date:	Sun, 13 Jun 2021	**Prob (F-statistic):**	2.20e-14
Time:	09:50:27	**Log-Likelihood:**	-60.432
No. Observations:	20	**AIC:**	124.9
Df Residuals:	18	**BIC:**	126.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.2183	2.549	0.478	0.638	-4.138	6.574
x1	10.1715	0.467	21.782	0.000	9.190	11.153

Omnibus:	12.388	**Durbin-Watson:**	1.903
Prob(Omnibus):	0.002	**Jarque-Bera (JB):**	2.388
Skew:	-0.249	**Prob(JB):**	0.303
Kurtosis:	1.382	**Cond. No.**	12.2

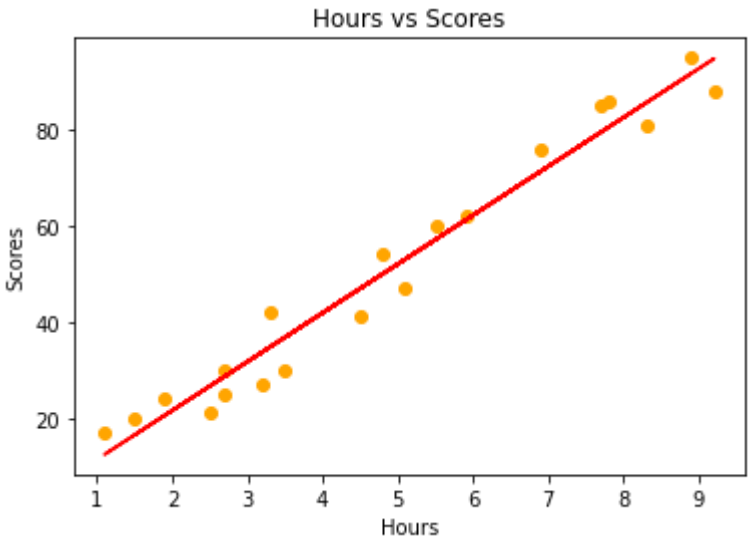
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

We can conclude from the above statistics that number of hours spent on study is highly significant at any chosen level of significance(alpha) and explains approximately 96% of variation in scores.

Plotting the Regression line

In [57]:

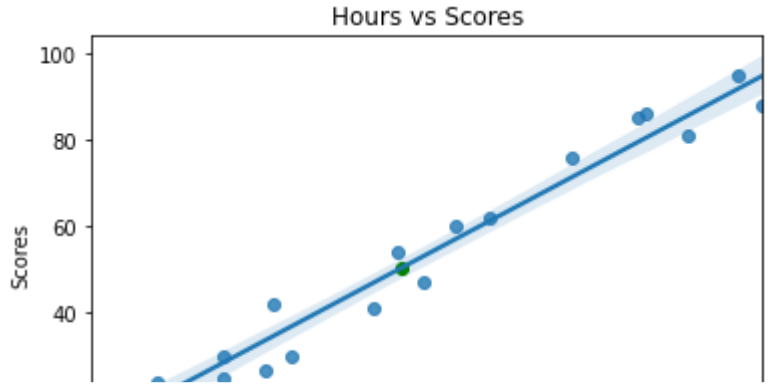
```
plt.scatter(x_train,y_train,c="orange")
plt.plot(x_train,regressor.predict(x_train), c="red")        # Point Estimates
plt.title('Hours vs Scores')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```



In [59]:

```
sns.regplot(x_train,y_train,fit_reg=True)
plt.scatter(np.mean(x_train),np.mean(y_train),color = 'green')
plt.title('Hours vs Scores')                                    # Confidence interval est
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```

```
C:\Users\worko\Ansh_programme\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From ve
rsion 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or mis
interpretation.
  warnings.warn(
```



Testing the model

In [39]:

```
y_pred = regressor.predict(x_test)        # Predicting the test data results on the basis o
y_pred
```

Out[39]:

```
array([[26.64701997],
       [76.4872762 ],
       [39.86994509],
       [87.67590515],
       [63.26435108]])
```

Predicting the score based on number of hours.

In [40]:

```
input = float(input('The number of hours studied is:'))
result = regressor.predict([[9.25]])
print('The predicted score is:',result)
```

```
The number of hours studied is:9.25
The predicted score is: [[95.3045158]]
```

In [41]:

```
from sklearn.metrics import mean_squared_error                    # in squared term
print('Mean Squared Error is:',mean_squared_error(y_test,y_pred))
```

```
Mean Squared Error is: 53.13035787401306
```

In [42]:

```
from sklearn.metrics import mean_absolute_error                   # in absolute term
print('Mean Absolute Error is:',mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error is: 6.424351078220741
```